

## **Rancang Bangun Sistem Keamanan dan Pengenalan Objek dalam Ruangan Sebagai Pengganti CCTV dengan Menggunakan Raspberry Pi**

**Nama** : Rasyid Sindu Prihantono  
**NRP** : 5110 100 052  
**Jurusan** : Teknik Informatika – FTIf ITS  
**Dosen Pembimbing I** : Ary Mahzaruddin Shiddiqi.,S.Kom.,  
M.Comp.Sc.  
**Dosen Pembimbing II** : Hudan Studiawan,S.Kom.,M.Kom.

### **ABSTRAK**

Pada saat ini berkembangnya teknologi telah mengubah pola pikir manusia, semakin banyaknya alat teknologi baru bermunculan semakin banyak ide yang dituangkan untuk membantu kehidupan manusia. Salah satu kebutuhan yang ada pada saat ini adalah sebuah sistem keamanan cerdas yang sederhana, mudah untuk di aplikasikan serta murah dalam segi biaya pembuatan maupun perawatan.

Seiring dengan banyaknya kebutuhan, maka diciptakanlah mini komputer Raspberry Pi yang berukuran sangat kecil dan dapat digunakan untuk berbagai macam kebutuhan fungsi, mulai dari multimedia sampai dengan *microcontroller* dan *server*. Tugas akhir ini dibuat untuk membantu mengatasi kebutuhan akan sebuah sistem keamanan ruangan yang mudah, murah dan efisien dibanding dengan penggunaan CCTV.

Dengan menggunakan Raspberry Pi yang telah ditambahi dengan sensor infra merah PIR untuk mendeteksi adanya gerakan dan pancaran suhu tubuh dari manusia atau objek yang memasuki ruangan, serta dilengkapi dengan kamera USB untuk melakukan ambil gambar manusia atau objek yang kemudian akan diproses dengan menggunakan Open-CV untuk menentukan manusia atau bukan, dan *Wi-Fi chipset* yang digunakan sebagai *access point* dan juga berfungsi melakukan pengenalan *MAC address client*.

Sistem ini juga akan dilengkapi dengan notifikasi berupa *e-mail* dan *update status* Twitter.

***Kata kunci: Sensor PIR, kamera USB, Pengenalan Objek Open-CV, Raspberry Pi access point.***

## **Design and Implementation Security Control Room System and Object Recognition to Replace Into CCTV using Raspberry Pi**

**Nama** : Rasyid Sindu Prihantono  
**NRP** : 5110 100 052  
**Jurusan** : Teknik Informatika – FTIf ITS  
**Dosen Pembimbing I** : Ary Mahzaruddin Shiddiqi.,S.Kom.,  
M.Comp.Sc.  
**Dosen Pembimbing II** : Hudan Studiawan,S.Kom.,M.kom.

### **ABSTRACT**

*In this era,the number of techonology development has changed the human's a way of thinking, there are many new device that can help human's life.*

*One of the most needed technology is a simple smart security system,not only easy to apply but also has low cost production and maintenance, because of that reason, Raspberry Pi has been made, has a small size and can be used for many purposes, from multimedia system to microcontroller or server. This project has been made to help people fullfil their needs about simple smart security system that can change CCTV function.*

*Using Raspberry Pi with some additional censor such as infra red PIR to detect motion and body heat radiation from human or object that passing through the device, it also using USB camera to take human picture or object. Then the result from previous process will be processed using Open-CV to detect the object is human or not*

*Wi-fi chipset that used as access point and also to recognize client MAC address. In this system also use some additional notification, such as e-mail and twitter status.*

**Keywords:** *Sensor PIR, kamera USB, Pengenalan Objek Open-CV, Raspberry Pi access point.*

*(Halaman ini sengaja dikosongkan)*

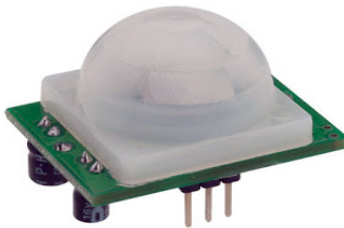
## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini dijelaskan beberapa hal mengenai teori yang berkaitan dengan sistem yang diimplementasikan. Hal ini ditujukan untuk memberikan gambaran secara umum terhadap sistem yang akan dibuat. Selain itu, hal tersebut berguna untuk menunjang pembuatan sistem sehingga kebutuhannya dapat diketahui.

#### **2.1. Sensor PIR**

Sensor PIR (*Passive Infra Red*) seperti pada Gambar 2.1 adalah sebuah sensor infra merah untuk mendeteksi adanya pancaran sinar infra merah. Sensor ini bersifat pasif atau tidak memancarkan infra merah tetapi hanya menerima pancaran radiasi infra merah dari luar, sebuah gerakan akan terdeteksi ketika sumber infra merah dengan suhu tertentu (misal: manusia) melewati sumber infra merah yang lain dengan suhu yang berbeda misalnya dinding, maka sensor akan membandingkan pancaran infra merah yang diterima setiap satuan waktu, sehingga jika ada pergerakan maka akan terjadi perubahan pembacaan sensor [1].

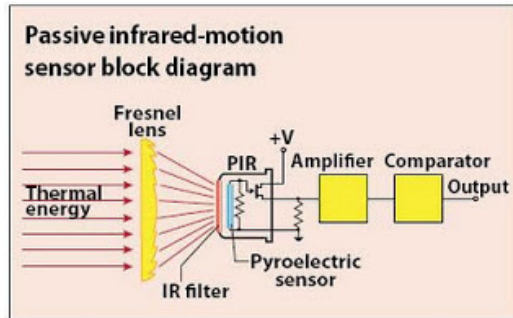


**Gambar 2.1 Sensor PIR**

Sensor PIR memiliki beberapa komponen seperti yang ditunjukkan pada Gambar 2.2:

- *Fresnel Lens*
- *IR Filter*
- *Pyroelectric sensor*

- *Amplifier*
- *Comparator*

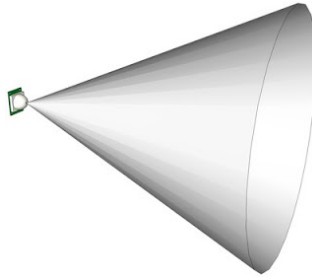


**Gambar 2.2 Komponen PIR**

Pada Gambar 2.2 ditunjukkan cara kerja pembacaan sensor PIR adalah pancaran infra merah masuk melalui lensa Fresnel dan mengenai sensor Pyroelectric, karena sinar infra merah mengandung energi panas maka sensor Pyroelectric akan menghasilkan arus listrik. Arus listrik inilah yang akan menimbulkan tegangan dan dibaca oleh sensor secara analog yang kemudian sinyal ini akan dikuatkan oleh penguat dan dibandingkan dengan *comparator* dengan referensi tegangan tertentu dan akan menghasilkan keluaran berupa 1-bit. Jadi PIR hanya akan menghasilkan keluaran berupa 0 atau 1, dimana 0 adalah ketika tidak ada pancaran sinar infra merah yang tertangkap oleh lensa dan 1 adalah ketika lensa menangkap pancaran sinar infra merah.

Sensor PIR didesain untuk hanya mendeteksi pancaran sinar infra merah dengan panjang gelombang 8-14 mikrometer, dengan kata lain pancaran diluar itu sensor tidak akan mampu untuk menangkapnya. Sedangkan manusia sendiri memiliki suhu badan rata-rata 32-33 derajat celcius yang akan menghasilkan panjang gelombang sekitar 9-10 mikrometer yang secara umum sensor PIR memang didesain untuk mendeteksi manusia.

Jarak pancaran PIR memiliki jangkauan yang bervariasi, tergantung dari karakteristik sensor PIR. Pada Gambar 2.3 ditunjukkan jarak atau area pancaran dari sensor PIR, biasanya jarak jangkauan PIR adalah sekitar 5 meter.



**Gambar 2.3 Pancaran sensor PIR**

## **2.2. Raspberry Pi**

Raspberry Pi adalah sebuah SBC (*single-board computer*) seukuran kartu kredit. Raspberry Pi telah dilengkapi dengan semua fungsi layaknya sebuah komputer lengkap, menggunakan SOC (*System-on-a-Chip*) ARM yang dikemas dan diintegrasikan di atas PCB (papan sirkuit). Raspberry Pi ini mampu bekerja layaknya komputer pada umumnya dengan kemampuan untuk menjalankan sistem operasi Linux dan aplikasinya seperti LibreOffice, multimedia (*audio* dan *video*), peramban *web*, ataupun *programming*. Raspberry Pi dapat menampilkan gambar ke TV/HDTV menggunakan koneksi HDMI ataupun TV standar menggunakan kabel RJ45 [2].

Beberapa sistem operasi yang dapat digunakan pada Raspberry Pi adalah:

- a. Raspbian OS
- b. Arch Linux ARM
- c. Raspbmc
- d. OpenELEC



**Gambar 2.4 Raspberry Pi**

Pada Gambar 2.4 adalah Raspberry Pi tipe B yang merupakan pengembangan dari Raspberry Pi tipe A yang telah dulu ada, pada Raspberry Pi tipe B ini memiliki beberapa contoh kegunaan atau fungsi sebagai berikut:

- a. *Media Server*.
- b. Dapat digunakan sebagai NAS (*Network Attached Storage*).
- c. *Print Server*.
- d. *Download Manager* untuk menghemat biaya listrik.
- e. Dapat digabungkan dengan berbagai macam sensor dan mikrokontroler seperti Arduino.
- f. *Home Automation* yaitu untuk mengontrol lampu, kipas angin, dan berbagai macam alat rumah tangga.
- g. *Server* untuk *Hosting Website*.
- h. Dapat digabung dengan beberapa Raspberry Pi untuk menghasilkan sebuah *cluster computing* untuk melakukan komputasi skala besar dengan biaya yang relative lebih murah.

## **2.3.ARP**

ARP adalah sebuah protokol dalam TCP/IP yang bertugas untuk melakukan resolusi alamat IP ke dalam alamat Media Access Control (*MAC address*).

Protokol ini berfungsi untuk memetakan *IP address* menjadi *MAC address* dan merupakan penghubung antara *datalink layer* dan *IO layer* pada TCP/IP. Semua komunikasi yang berbasis



*Ethernet* menggunakan protokol ARP ini, pada dasarnya adalah setiap komputer atau *device* yang akan berkomunikasi pasti akan melakukan transaksi atau tukar menukar informasi terkait antara IP dan MAC *address*, setiap transaksi akan disimpan di dalam *cache operating system* anda [3].

Paket Informasi pada bagian ARP dapat dipisahkan menjadi 2 type tergantung pada Jenis Reciever yang diberikan/diijinkan pada jaringan, yaitu:

- a. Broadcast: Alamat MAC address yang dituju ditampilkan / dikirim ke semua penerima dalam jaringan LAN saat Switch Jaringan menerima penghubungan/ konektivitas perangkat.
- b. Non-Broadcast: Hanya beberapa Host yang telah ditentukan dapat menerima paket pengiriman.

Jenis dari paket ARP juga dapat dibagi menjadi 2 jenis berdasarkan fungsi:

- a. ARP Request: digunakan untuk mengakses MAC address dan mengelolanya melalui IP address yang terbaca/terdaftar di dalam jaringan LAN.
- b. ARP Reply: digunakan untuk menginformasikan ke suatu Host dalam jaringan mengenai bagian localhost dari IP address dan MAC Address

```
arp -n 10.151.63.212 | awk ' 10.151.63.212 {print $3} |  
tail -n 1
```

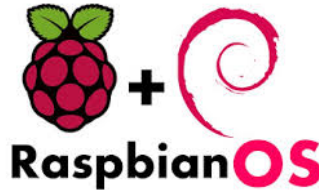
**Gambar 2.5 Contoh script ARP**

Pada Gambar 2.5 adalah contoh *script* sederhana untuk menjalankan *command shell* dari ARP untuk IP 10.151.63.212 dan untuk menghasilkan *output* berupa MAC *address* dari *device* dengan alamat IP 10.151.63.212.

## **2.4.Raspbian OS**

Pada Gambar 2.6 ditunjukkan logo dari sistem operasi Raspbian yang merupakan sebuah sistem operasi berbasis Linux *distro* Debian yang dapat dioptimalkan untuk penggunaan komputer mini Raspberry Pi [4]. Sistem operasi ini memiliki

beberapa *program* standard dan beberapa *program* pembantu untuk dapat menjalankan perangkat keras dari komputer mini Raspberry Pi ini.



**Gambar 2.6 Lambang Raspbian OS**

Dalam sistem operasi ini sudah lebih lengkap daripada sistem yang murni digunakan di komputer pada umumnya karena memiliki lebih dari 350.00 paket dan *library pre-compiled* yang tersaji dalam bentuk format yang mudah untuk diinstalasi pada Raspberry Pi.

## **2.5.Raspberry Pi GPIO Pin**

*General Purpose Input Output* atau yang biasa disebut *socket* GPIO adalah pin yang dimana berfungsi sebagai pin *socket input/output* dari perangkat keras dan dapat dikendalikan melalui perangkat lunak. *Socket* GPIO dapat digunakan untuk menghubungkan beberapa perangkat tambahan seperti sensor infra merah, layar LCD, lampu LED ataupun untuk menambahkan perangkat Arduino. Kemampuan dari *socket* GPIO adalah antara lain sebagai berikut:

- a. Pin GPIO dapat dikonfigurasi untuk menerima *input* dan mengirim *output*
- b. Pin GPIO dapat diaktifkan dan dinonaktifkan
- c. Nilai *input* dari GPIO adalah 1 dan 0, dimana 1 = *high* dan 0 = *low*
- d. GPIO dapat dipakai beberapa sensor dengan menambahkan perangkat keras yang disebut *Cobler*, alat ini berfungsi untuk menambahkan dan menghubungkan *pin* GPIO yang ada pada Raspberry Pi dengan *pin* yang



- n. Pin 14 berfungsi sebagai pin *Ground*
- o. Pin 15 berfungsi sebagai pin GPIO 22
- p. Pin 16 berfungsi sebagai pin GPIO 23
- q. Pin 17 berfungsi sebagai pin *power* dengan daya 3,3 V
- r. Pin 18 berfungsi sebagai pin GPIO 24
- s. Pin 19 berfungsi sebagai pin GPIO 10
- t. Pin 20 berfungsi sebagai pin *Ground*
- u. Pin 21 berfungsi sebagai pin GPIO 9
- v. Pin 22 berfungsi sebagai pin GPIO 25
- w. Pin 23 berfungsi sebagai pin GPIO 11
- x. Pin 24 berfungsi sebagai pin GPIO 8
- y. Pin 25 berfungsi sebagai pin *Ground*
- z. Pin 26 berfungsi sebagai pin GPIO 7

## 2.6. Fswebcam

Fswebcam adalah suatu aplikasi *webcam* atau modul kamera yang dapat melakukan ambil gambar melalui media tertentu seperti kamera atau *webcam*. Aplikasi ini dapat digunakan untuk mereduksi *noise* yang ditimbulkan oleh kamera saat melakukan perintah ambil gambar dan selain itu aplikasi ini dapat melakukan kompresi ukuran gambar yang dihasilkan berdasarkan format yang diinginkan seperti PNG atau JPEG.

Hasil dari proses ambil gambar ini dapat diolah lagi sesuai dengan kebutuhan dan keinginan pengguna, seperti pengaturan resolusi gambar yang diinginkan, info gambar yang ingin ditampilkan pada hasil akhir tangkapan gambar pada kamera, selain itu Fswebcam dapat melakukan perintah ambil gambar secara berurutan dan dapat disesuaikan waktu pengambilannya. Gambar 2.8 adalah contoh hasil foto dengan menggunakan Fswebcam dan camera Logitech tipe C-170.

Untuk pengambilan gambar dengan menggunakan Fswebcam kita dapat menggunakan perintah dari *command shell* yang kemudian kita dapat menambahkan pengaturan atau *parameter* apa saja yang kita inginkan untuk proses pengambilan gambar dengan Fswebcam.



**Gambar 2.8 Contoh hasil foto dengan Fswebcam**

Gambar 2.8 adalah contoh hasil dari proses pengambilan gambar dengan menggunakan Fswebcam tanpa melakukan penambahan *parameter* apapun di dalam *command shell* program Fswebcam.

## 2.7. TweetPony

TweetPony adalah sebuah *library* API yang terdapat pada Python, berfungsi untuk mengakses sosial media Twitter dengan menggunakan bahasa pemrograman Python [6]. Dengan menggunakan TweetPony pengguna dapat melakukan beberapa fungsi standar pada Twitter seperti *update* status Twitter dan *upload* gambar.

Sebelum menggunakannya, pengguna diharuskan mendaftarkan terlebih dahulu aplikasi yang ingin menggunakan hak akses API dari Twitter guna mendapatkan *consumer\_key* dan *access\_token*, keduanya adalah syarat penting agar kita dapat mendapatkan hak akses dari Twitter.

## 2.8. Dnsmasq

Dnsmasq adalah sebuah aplikasi yang menyediakan layanan sebagai DNS *cacher* dan DHCP *server*, sebagai DNS *server*, Dnsmasq dapat melakukan *cache* DNS *query* untuk meningkatkan kecepatan dari koneksi untuk mengakses situs-situs sebelumnya yang telah diakses dan sebagai DHCP *server*, Dnsmasq dapat digunakan untuk menyediakan IP *local* dan melakukan routing kekomputer yang ada pada jaringan LAN [7].

Dnsmasq sangat ringan dan sangat mudah untuk dikonfigurasi, didesain untuk jaringan yang lebih kecil dengan jumlah komputer tidak lebih dari 50 dalam satu jaringan.

Untuk melakukan instalasi dnsmasq kita dapat menggunakan perintah `apt-get install -y dnsmasq`, lalu kita masuk ke `/etc/dnsmasq.conf` untuk melakukan konfigurasi pada dnsmasq atau dapat menggunakan shell *command*.

```
sudo dnsmasq -d -k --interface=wlan0 --dhcp-range = 172.63.0.2,172.63.0.10 --conf-file = /dev/null
```

**Gambar 2.9 Shell command untuk Dnsmasq**

Pada Gambar 2.9 merupakan contoh untuk mengaktifkan Dnsmasq melalui shell *command*, berikut penjelasan:

- `--interface= wlan 0`  
Menggunakan wlan0 atau *wi-fi chipset* sebagai *interface*.
- `Dhcp-range = 172.63.0.2,172.63.0.10`  
Membagi alamat IP pada rentang IP 172.63.0.2 sampai dengan 172.63.0.10

Dengan menjalankan perintah seperti pada Gambar 2.10 maka *client* akan mendapat IP *address* pada rentan IP 172.63.0.2 sampai dengan 172.63.0.10.

## 2.9. Hostapd

Hostapd adalah sebuah aplikasi yang dapat berjalan secara *daemon* atau dibelakang layar yang berfungsi untuk menjadikan Raspberry Pi atau *device* kita sebagai *router access point* [8], untuk menginstallnya kita dapat menggunakan perintah `apt-get install hostapd`.

```
Interface=wlan0
Ssid=dududh-wifi
Channel = 6
Hw_mode = g
Ieeee80211n = 1
```

```
Wpa = 2
Wpa_passphrase = SemogaBerhasil
```

**Gambar 2.10 Konfigurasi pada dudud-wifi.conf**

Pada Gambar 2.10 ditunjukkan pengaturan dari *file* dudud-wifi.conf, yaitu:

1. Interface = wlan0
2. Menggunakan wlan0 sebagai *interface*.
3. SSID = dudud-wifi
4. Memberikan nama pada *access point* yang akan kita buat.
5. Channel = 6
6. Set *WiFi channel*, dapat diubah mulai dari 1-11.
7. Hw\_mode = g
8. Set *access point hardware* mode ke 802.11g
9. Ieee80211n
10. *Allow device* dengan spesifikasi yang sesuai dengan *ieee80211n*
11. Wpa = 2
12. *Enable wpa2*
13. Wpa-passphrase = SemogaBerhasil
14. Merupakan *password* untuk mengakses kedalam *access point* kita.

## 2.10. Open-CV

OpenCV adalah suatu *library* gratis yang dikembangkan oleh *developer-developer* Intel Corporation. *Library* ini terdiri dari fungsi-fungsi *computer vision* dan API (*Application Programming Interface*) untuk *image processing* yang *high level* maupun *low level* dan sebagai optimasi aplikasi *realtime*. OpenCV [9] sangat disarankan untuk programmer yang akan berkecukupan pada bidang *computer vision*, karena *library*nya mampu membuat aplikasi yang handal di bidang *digital vision* dan mempunyai fitur yang mirip dengan cara pengolahan visual pada manusia.

Berikut ini adalah beberapa fitur pada *library* OpenCV:

1. Manipulasi data gambar (alokasi memori, melepaskan memori, setting serta konversi gambar)
2. Image/Video I/O (Bisa menggunakan camera yang sudah didukung oleh *library* ini)
3. Manipulasi matrix dan vektor serta terdapat juga *routines* linier algebra (*products*, *solvers*, *eigenvalues*, SVD)
4. *Image processing* dasar (*filtering*, *edge detection*, pendeteksian tepi, *sampling* dan *interpolasi*, konversi warna, operasi *morfologi*, *histograms*, *image pyramids*)
5. Analisis struktural
6. Kalibrasi kamera
7. Pendeteksian gerak
8. Pengenalan objek (hewan, manusia, tumbuhan dll)
9. Basic GUI (*Display* gambar/video, mouse/keyboard kontrol, *scrollbar*)
10. *Image Labelling* (*line*, *conic*, *polygon*, *text drawing*)

Dengan memanfaatkan *library* pada openCV yaitu fungsi HOG (Histogram Oriented of Gradient) *descriptor*, maka suatu image dapat dideteksi objek tersebut atau manusia atau bukan. HOG *descriptor* sendiri merupakan suatu fitur dari gambar yang digunakan untuk menghitung vektor gradien pada area tertentu sehingga dihasilkan output berupa vektor yang nantinya diklasifikasi oleh *support vector machine* [10].



## **BAB III**

### **PERANCANGAN PERANGKAT LUNAK**

Perancangan merupakan bagian penting dalam pembangunan perangkat lunak yang berupa perencanaan-perencanaan secara teknis mengenai aplikasi yang dibuat. Bab ini secara khusus akan menjelaskan perancangan sistem yang dibuat dalam Tugas Akhir ini. Penjelasan mencakup deskripsi umum aplikasi hingga perancangan proses, alur dan implementasinya.

#### **3.1.Deskripsi Umum Sistem**

Pada tugas akhir ini akan dibangun suatu sistem pemantau ruangan dan identifikasi objek manusia berbasis mini komputer Raspberry Pi sebagai pengganti CCTV. Sistem ini dirancang dengan studi kasus di dalam suatu ruangan atau rumah yang dalam kondisi kosong atau tak tidak berpenghuni yang lemah terhadap pengawasan orang atau penyusup, sistem ini akan mengambil data dari inputan sensor PIR (*Passive Infra Red*) yang bertugas untuk menangkap panas suhu tubuh seseorang yang sedang melewati suatu ruangan atau rumah yang telah diawasi oleh rangkain alat ini.

Sensor akan bernilai logika 1 atau 0 dimana kondisi 1 adalah ketika sensor menangkap gerakan dan panas suhu tubuh dari manusia yang melewatinya sedangkan kondisi 0 adalah dimana tidak ada gerakan dan suhu tubuh yang melewati sensor. Mini komputer Raspberry Pi berfungsi sebagai pengendali utam dari semua perangkat yang ada dan dihubungkan menjadi satu, baik itu berupa sensor PIR, USB kamera, USB Hub, dan *Wi-Fi Chipset*.

Di sini seluruh perangkat akan mendapatkan suplai daya listrik dari 2 sumber daya yang terpisah, yang pertama adalah berasal dari *adapter* yang langsung akan terhubung dengan Raspberry Pi dan yang kedua adalah USB Hub dengan *adaptor* yang akan tersambung dari listrik rumah yang kemudian akan disambungkan ke Raspberry Pi. Hal ini diperlukan agar suplai daya listrik ke USB kamera tidak hanya mengandalkan suplai

listrik dari Raspberry Pi saja dikarenakan keterbatasan suplai daya listrik pada Raspberry Pi yang akan mengakibatkan USB kamera dan *Wi-fi chipset* tidak akan berjalan, dimana hasil proses pengambilan gambar dari kamera akan diolah di dalam Raspberry Pi untuk menentukan apakah itu manusia atau objek selain manusia [11]. Untuk perantara ke jaringan internet maka penulis menggunakan jaringan LAN yang tersedia di Laboratorium GCL (*Grid Computing laboratory*) yang telah dikonfigurasi sebelumnya agar mampu mengubah Raspberry Pi menjadi *access point* dan *client* dapat menggunakan akses internet melalui Raspberry Pi *access point* [12], dimana *MAC address client* yang terhubung akan dikenali oleh Raspberry Pi *access point* sebagai penyusup atau anggota keluarga yang kemudian mengirim pesan notifikasi jika ada penyusup atau anggota keluarga yang berada di dalam ruangan atau rumah [12].

Notifikasi berupa status Twitter oleh akun yang telah mendaftarkan sistem ini ke API Twitter Development sebelumnya, dalam kasus ini akun yang digunakan adalah akun pribadi dari penulis.

Untuk dapat menggunakan API Twitter Development, kita harus mendaftarkan aplikasi kita terlebih dahulu ke Twitter Development untuk mendapatkan *access\_key* dan *secret\_token*, kedua elemen ini adalah sebuah barisan angka dan huruf unik sebagai alat *validasi* dari API Twitter Development yang telah menyetujui aplikasi kita untuk menggunakan API tersebut, *Access\_key* dan *Secret\_token* dari setiap pengguna berbeda.

Sedangkan untuk menangkal serangan dari *ARP Spoofing* dan *ARP poisoning*, maka sistem akan melakukan *binding IP address* dan *MAC address* kepada *client* yang terhubung ke dalam *access point*.

### 3.2. Arsitektur Umum Sistem

Pada Gambar 3.1 dijabarkan alur kerja sistem adalah sebagai berikut:

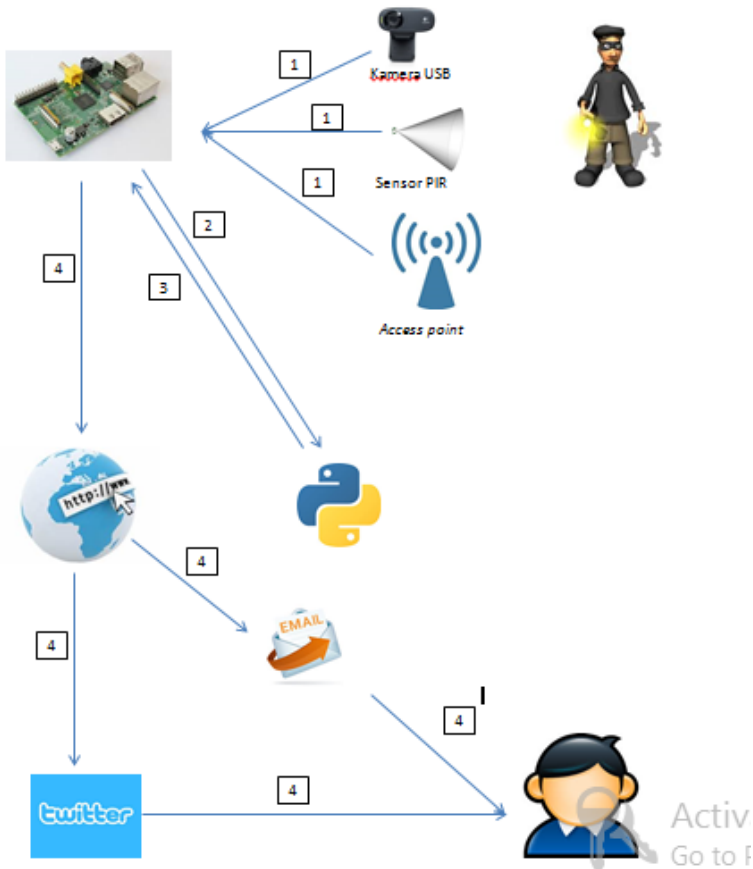
1. Dengan cara mendeteksi sensor panas tubuh dan gerakan dari seseorang atau objek yang memasuki dan melewati

ruangan tersebut, dengan menggunakan sensor infra merah yang sudah di integrasikan melalui *socket* GPIO dengan mini komputer Raspberry Pi. Setelah sensor menangkap pancaran sinar infra merah dari suhu tubuh manusia atau objek yang tidak sengaja tertangkap dalam radius pancaran PIR, selain menggunakan sensor infra merah Raspberry Pi yang telah dimodifikasi menjadi *access point* ini akan melakukan *scanning* *MAC address client* yang telah tersambung kedalam *access point* Raspberry Pi yang kemudian akan dicocokkan dengan *list* yang telah ada.

2. Terdapat dua kondisi dalam proses *scanning* ini, kondisi yang pertama adalah dimana ketika dikenalnya *MAC address* dari *client* yang terhubung dengan isi *list* maka akan dikenali sebagai anggota keluarga yang kemudian akan dilakukan proses pengambilan gambar dengan menggunakan kamera yang telah diintegrasikan dengan Raspberry Pi setelah proses pengambilan gambar selesai akan dilakukan proses *update status* Twitter disertai hasil foto tersebut. Kondisi kedua adalah ketika dimana ketika melakukan pengecekan *MAC address* tidak dikenali di dalam *list* yang ada, yang kemudian akan dikenali sebagai penyusup dan disimpan dalam *directory* tersendiri.
3. Setelah melakukan pengambilan gambar penyusup, hasil gambar tersebut kemudian diolah dengan *script* bahasa pemrograman Python untuk mendeteksi apakah objek tersebut adalah manusia atau bukan [13].
4. Ketika diketahui objek dari pengambilan gambar penyusup tersebut adalah manusia maka sistem akan secara otomatis akan melakukan notifikasi berupa *update status* Twitter disertai gambar dan notifikasi berupa *e-mail* kepada *user* agar penghuni dapat segera mengetahui bahwa diruangan tersebut ada penyusup, namun ketika hasil pengolahan gambar tersebut menyatakan objek

bukan manusia maka sistem akan kembali kedalam keadaan semula atau *ready*.

Rancangan arsitektur sistem ditunjukkan seperti pada Gambar 3.1 dibawah ini.



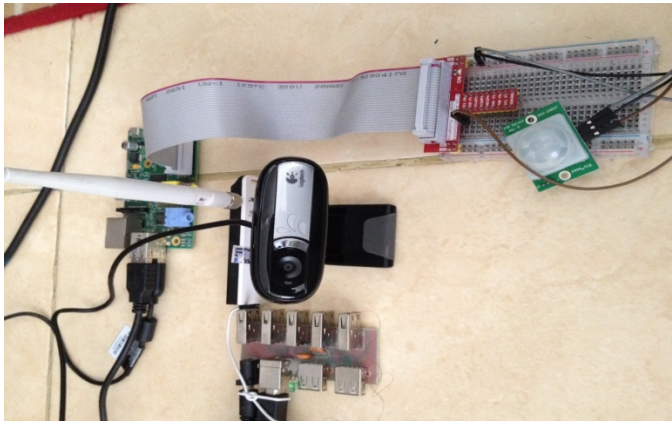
Gambar 3.1 Arsitektur sistem secara umum

### 3.3. Perancangan Perangkat Keras

Pada Gambar 3.2 ditunjukkan dari hasil rangkaian perangkat keras yang akan digunakan pada sistem pengawas ruangan

berbasis mini komputer Raspberry Pi, berikut penjabaran dari perangkat keras yang akan digunakan:

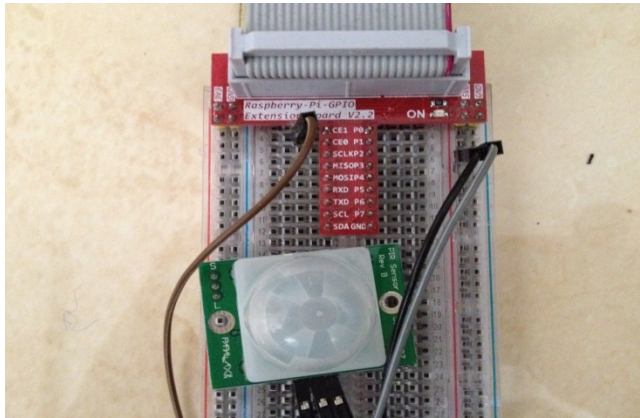
- 1 buah mini komputer Raspberry Pi.
- 1 buah sensor PIR infra merah.
- 1 buah t-cobler.
- 1 buah *webcam* Logitech c 170.
- 1 buah SDHC V-Gen 8gb.
- 3 buah kabel GPIO pelangi.
- 1 buah kabel GPIO *Ribbon*.
- 1 buah *beard board* 400 pin.
- 1 buah *Wi-Fi chipset* TP-Link-WN722N.
- 1 buah kabel LAN.
- 1 buah *USB Hub with adapter* 5v 1000Ma.
- 1 buah adaptor 5V 1000Ma.
- 1 buah monitor *support* HDMI.
- 1 buah kabel HDMI



**Gambar 3.2 Hasil rangkaian perangkat keras**

Pada Gambar 3.3 ditunjukkan rangkaian dari *socket* GPIO pada komputer mini Raspberry Pi yang telah digabung dengan *beard board* dengan menggunakan kabel *ribbon* dan *t-cobler*. Meskipun pada mini komputer Raspberry Pi telah dilengkapi dengan *socket* GPIO yang dapat langsung dipasangkan dengan

sensor infra merah PIR, tetapi penulis memilih menggunakan tambahan *beard board* dan kabel *ribbon* karena untuk memudahkan ketika kita akan menambahkan beberapa sensor dan memudahkan dalam membaca struktur *socket* GPIO.



**Gambar 3.3 Rangkaian breadboard**

Pada rangkaian di atas mini komputer Raspberry Pi berfungsi sebagai pusat dari semua perangkat keras dan sistem yang akan berjalan, baik itu berupa sensor, *webcam*, maupun *Wi-Fi chipset* yang dipasangkan pada Raspberry Pi. Sensor infra merah berfungsi sebagai penangkap panas suhu tubuh manusia atau objek yang sedang melewati suatu ruangan yang sedang diawasi oleh rangkaian dari sistem pengawas ruangan berbasis mini komputer Raspberry Pi, *Wi-Fi chipset* berfungsi sebagai sarana untuk menjadi *access point* yang berfungsi selain sebagai *access point* juga berfungsi untuk pengenalan *MAC address client* yang kemudian akan diproses untuk mengenali anggota keluarga atau bukan.

Sensor akan bernilai logika 1 dan 0, bernilai 1 ketika ada gerakan dan pancaran suhu tubuh manusia yang tertangkap pada radius pancaran PIR dan akan bernilai 0 ketika tidak ada nya gerakan dan pancaran suhu tubuh manusia atau objek dalam radius pancaran PIR.

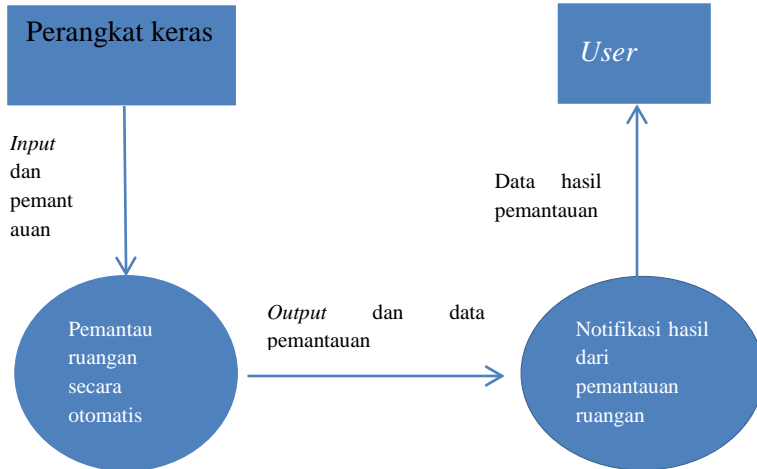
Seluruh perangkat yang penulis rangkai mendapatkan suplai daya listrik dari listrik rumah ditambah dengan USB *Hub* untuk memenuhi kebutuhan listrik dari *Wi-Fi chipset* yang dipasang pada mini komputer Raspberry Pi. Untuk perantara internet digunakanlah jaringan LAN yang terdapat pada Laboratorium GCL yang sudah dikonfigurasi dengan DNS dan IP *public* sehingga untuk mengakses internet tidak diperlukan *proxy*, hal ini dilakukan untuk memudahkan mengakses internet bagi sistem dan *client* yang akan terhubung kedalam *access point* Raspberry Pi dan juga untuk memudahkan dalam pengiriman berkas file foto dan *update status* Twitter.

### 3.4. Perancangan Diagram Alir Data *Level 0*

Diagram alir data *level 0* menggambarkan fungsionalitas sistem beserta actor yang terlibat. Seperti yang terlihat pada Gambar 3.4 ditunjukkan bahwa terdapat satu pengguna yang menggunakan sistem ini. Pengguna dapat melihat hasil dari gambar yang ditangkap oleh sistem pengawas ruangan ini dengan menggunakan *handphone* atau *laptop* secara *online* dengan cara membukan akun *e-mail* atau Twitter yang sudah dikonfigurasi sebelumnya di dalam sistem pengawas ruangan ini. Hasil tangkapan gambar dari sistem pengawas ini akan berjalan secara otomatis atau bersifat *realtime*.

Sistem akan diawali dengan *input* hasil dari pendeteksian pancaran panas suhu tubuh manusia yang tertangkap oleh sensor infra merah, yang bernilai 1 atau 0. Yang kemudian dilanjutkan dengan pengecekan *MAC address client* yang tersambung kedalam *access point* mini komputer Raspberry Pi, dari hasil kedua kondisi di atas akan *men-trigger* kamera yang telah terhubung dengan mini komputer Raspberry Pi untuk melakukan perintah ambil gambar dengan konfigurasi gambar yang sudah disesuaikan sebelumnya. Ada dua kondisi dari hasil pengecekan *MAC address client* yang tersambung kedalam *access point* Raspberry Pi, kondisi pertama adalah ketika *MAC address client* dikenali oleh *access point* Raspberry Pi yang kemudian akan menghasilkan *output* berupa nama anggota keluarga, kondisi

kedua adalah ketika *MAC address client* tidak dikenali oleh *access point* Raspberry Pi yang akan menghasilkan *output* berupa penyusup, dari hasil kedua akan diolah kembali dengan Raspberry Pi untuk menentukan objek tersebut adalah manusia atau bukan.



**Gambar 3.4 Perancangan diagram sistem level 0**



## **BAB IV IMPLEMENTASI**

Setelah melalui proses analisis dan perancangan perangkat lunak, dilakukan implementasi sistem. Bab ini membahas tentang proses implementasi pada sistem.

### **4.1. Lingkungan Implementasi**

Dalam proses implementasi sistem ini, digunakan beberapa perangkat pendukung untuk memenuhi kebutuhan sistem yang terdiri dari perangkat keras dan perangkat lunak.

#### **4.1.1. Lingkungan Implementasi Perangkat Keras**

Perangkat keras yang digunakan dalam pengembangan sistem ini adalah komputer/*notebook* dan mini komputer Raspberry Pi. Spesifikasi dari perangkat-perangkat tersebut adalah sebagai berikut :

- *Notebook* Dell Inspiron N 4010, Intel(R) Core(TM) i3 CPU M350 @2.27Ghz (4CPUs), ~2.3Ghz dengan memori 4096MB DDR2.
- Raspberry Pi model B versi UK dengan SDHC V-Gen 8gb.

#### **4.1.2. Lingkungan Implementasi Perangkat Lunak**

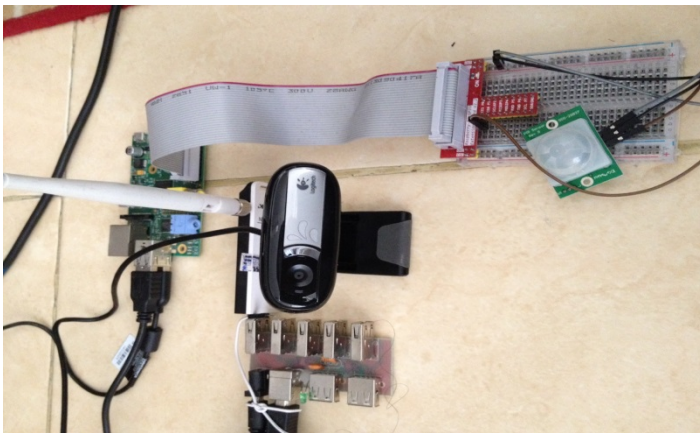
Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem adalah sebagai berikut:

- Micorosft Windows 8 64-bit sebagai sistem operasi
- Raspbian wheezy *release date* 2013-09-25 sebagai sistem operasi pada mini komputer Raspberry Pi.
- Bahasa pemrograman Python untuk mengimplementasikan aplikasi dan mengintegrasikan perangkat keras yang tersambung dengan komputer mini Raspberry Pi.
- PuTTY sebagai aplikasi untuk melakukan *remote control* terhadap mini komputer Raspberry Pi.

- Tmux sebagai alat untuk membagi *screen* pada terminal Linux untuk menjalankan beberapa aplikasi secara bersama atau berurutan.

#### 4.2.Implementasi Perangkat Keras

Pada Gambar 4.1 ditunjukkan pembuatan implementasi perangkat keras ini diawali dengan sebuah *prototype* untuk menguji apakah perangkat keras dapat berfungsi sebagai sistem pengawas ruangan. Pada sistem ini, perangkat keras yang digunakan adalah sebagai berikut:



**Gambar 4.1 Implementasi perangkat keras**

- 1 buah mini komputer Raspberry Pi.
- 1 buah sensor PIR infra merah.
- 1 buah t-cobler.
- 1 buah *webcam* Logitech c 170.
- 1 buah SDHC V-Gen 8gb.
- 3 buah kabel GPIO pelangi.
- 1 buah kabel GPIO *Ribbon*.
- 1 buah beard board 400 pin.
- 1 buah *Wi-Fi chipset* TP-Link-WN722N.
- 1 buah kabel LAN.

- 1 buah USB *Hub with adapter* 5v 1000Ma.
- 1 buah adaptor 5V 1000Ma.
- 1 buah monitor *support* HDMI.
- 1 buah kabel HDMI.
- 1 buah *notebook* Dell Inspiron n 4010

Pengujian fungsi perangkat sensor melibatkan semua komponen perangkat keras yang ada pada rangkaian. *Prototype* perangkat keras pada tugas akhir ini menggunakan sebuah mini komputer Raspberry Pi model B yang dilengkapi dengan kartu SD V-Gen berkapasitas 8GB dan sudah terisi dengan sistem operasi Raspbian Wheezy, sensor infra merah PIR, USB kamera Logitech bertipe C-170 dengan kemampuan optik *5megapixel*, *Wi-Fi chipset* TP-Link-WN722N dan USB *hub* dengan adaptor berkapasitas 5V 1000MA sebagai penambah daya. Mini komputer Raspberry Pi berfungsi sebagai pusat dari semua sistem dari perangkat dan pengontrol algoritma dan pengolahan data.

P1			
<50mA	3V3		1 2
BCM GPIO00/02	SDA0/1	8	3 4
BCM GPIO01/03	SCL0/1	9	5 6
BCM GPIO04		7	7 8
	GND		9 10
BCM GPIO17		0	11 12
BCM GPIO21/27		2	13 14
BCM GPIO22		3	15 16
<50mA	3v3		17 18
BCM GPIO10	SPIMOSI	12	19 20
BCM GPIO9	SPIMOSO	13	21 22
BCM GPIO11	SPI SCLK	14	23 24
	GND		25 26
P5			
<50mA	3V3		2 1
BCM GPIO29	SCL0	18	4 3
BCM GPIO31		20	6 5
	GND		8 7

**Gambar 4.2 Diagram socket GPIO Raspberry Pi**

Pada Gambar 4.2 ditunjukkan diagram *socket* GPIO untuk menghubungkan mini komputer Raspberry Pi dengan sensor infra merah PIR, dibutuhkan beberapa kabel GPIO pelangi untuk menghubungkan *socket* dari modul sensor infra merah PIR ke mini komputer Raspberry Pi. Penulis menambahkan perangkat keras *T-Cobler* guna memudahkan pemasangan antara GPIO

dengan sensor infra merah yang dirangkai di dalam Beard Board 400 pin, fungsi utama dari *board* ini adalah memudahkan ketika akan ditambahkan sensor-sensor tambahan kedalam rangkaian. Pada sensor infra merah PIR terdapat 3 *pin socket* utama agar sensor dapat berjalan pada sistem, ketiga *pin* tersebut adalah *VCC*, *GND*, dan *digital pin*.

*Pin* pertama adalah *VCC* yaitu *pin power* yang harus diberi daya sebesar 5 V, maka untuk memperoleh sumber daya tersebut *pin* harus ditancapkan pada Beard Board dengan *T-Cobler* yang bertuliskan 5V0 yang secara langsung akan terhubung dengan *pin* GPIO nomor 2 pada Raspberry Pi, lalu *pin* kedua adalah *pin GND* atau *ground* pada sensor infra merah PIR ditancapkan pada Beard Board dengan *T-Cobler* bertuliskan *GND* yang akan secara langsung terhubung dengan *pin* GPIO nomor 6 pada Raspberry Pi dan *pin* yang terakhir adalah *pin digital* yang akan kita tancapkan pada Beard Board dengan *T-Cobler* bertuliskan CE1 yang akan terhubung dengan *socket* GPIO7 pada *socket* GPIO Raspberry Pi nomor 26.

Untuk *Wi-Fi chipset* dipasang terlebih dahulu ke rangkaian USB *Hub* dengan adaptor berkapasitas 5 V 1000Ma yang telah dihubungkan kedalam USB *Hub* yang tersedia pada Raspberry Pi, diperlukannya USB *Hub* dengan adaptor adalah untuk mengatasi kurangnya sumber daya listrik pada Raspberry Pi yang terbagi untuk beberapa perangkat keras yang terpasang seperti USB kamera dan sensor infra merah PIR.

Pada percobaan awal dilakukan pada saat ada seseorang yang melintas melewati sensor infra merah tersebut untuk melihat apakah sensor infra merah PIR bekerja dengan baik terhadap tangkapan panas suhu tubuh manusia. Rangkaian diletakkan dengan jarak 1-2 meter dari seseorang yang berada dalam cangkupan pancaran sensor infra merah PIR, kemudian dipasangkannya tambahan perangkat keras USB *Hub* dengan adaptor agar ketika sistem berjalan tidak terjadi kekurangan daya listrik pada perangkat keras dan Raspberry Pi.

### 4.3. Implementasi pada Perangkat Lunak

Pada subbab-subbab berikut akan dijelaskan bagaimana implementasi perangkat lunak pada mini computer Raspberry Pi agar semua perangkat yang tersambung dapat bekerja dengan baik dan sesuai dengan susunan yang ditentukan agar system pengawas ruangan ini dapat berjalan dengan baik dan benar.

#### 4.3.1. Implementasi Sensor PIR

Implementasi sensor infra merah PIR pada mini komputer Raspberry Pi memiliki dua kondisi, kondisi pertama adalah ketika ada panas suhu tubuh manusia yang tertangkap oleh sensor infra merah PIR maka *current\_state* akan menghasilkan nilai 1 dan *previous\_state* akan bernilai 0. Kondisi yang kedua adalah dimana ketika tidak ada *trigger* yang masuk kedalam sensor infra merah PIR maka *current\_state* dan *previous\_state* akan bernilai sama sama 0. Sensor infra merah PIR akan dihubungkan ke *pin socket* GPIO7 pada Raspberry Pi, seperti pada Gambar 4.3 di bawah ini.

```

Import RPi.GPIO as io
pir_pin = 7 #inisialisasi pin GPIO pada
Raspberry Pi
io.setup(pir_pin, io.IN)

current_state = 0
previous_state = 0
if current_state==1 and previous_state==0:
#kondisi ketika PIR menerima trigger
print " aktif"
elif current_state==0 and previous_state==0:
#kondisi dimana PIR tidak mendapatkan trigger
print "tidak aktif"

```

**Gambar 4.3 Inisialisasi sensor PIR**

1. Pada Gambar 4.3 ditunjukkan potongan *syntax program* untuk melakukan inisialisasi sensor infra merah PIR ke *socket* GPIO pada Raspberry Pi.

2. Inisialisasi *socket* yang digunakan adalah sebagai perantara sinyal digital yang dikirim oleh sensor infra merah PIR ke mini komputer Raspberry Pi.
3. Jika sensor infra merah PIR menerima *trigger* maka akan menampilkan *output* “ aktif “ dan ketika tidak ada *trigger* akan menampilkan *output* “ tidak aktif”.

### 4.3.2. Implementasi Kamera

Modul kamera adalah alat yang berperang penting dalam sistem ini, karena berfungsi sebagai alat untuk menangkap gambar dari seseorang atau objek yang sebelumnya telah terdeteksi terlebih dahulu oleh sensor infra merah PIR pada Raspberry Pi, berikut adalah proses implementasi modul kamera pada Raaspberry Pi:



**Gambar 4.4 Kamera USB Logitech C-170**

Gambar 4.4 adalah modul kamera yang digunakan penulis untuk dihubungkan dengan mini komputer Raspberry Pi, dengan merek Logitech seri C-170 yang memiliki spesifikasi resolusi optik sebesar 5 *megapixel* dapat mengambil gambar dengan resolusi *high definition* sekitar 1280 x 720. Proses pengambilan gambar akan dijelaskan pada *syntax* program dibawah ini:

```
Import subprocess as sb
grap_cam = sb.Popen("sudo fswebcam -p YUYV -d
/dev/video0 -r 480x640 /home/pi/anggota/%m-%d-
%y-%H%M.jpg", shell=True)
#memanggil program fswebcam melalui subprocess call
img_name = today_date.strftime('%m-%d-%y-%H%M')
```

```
#format penamaan pada hasil foto
img_path = "/home/pi/anggota/{0}.jpg".format(img_name)
#media penyimpanan hasil foto
```

#### **Gambar 4.5 Syntax proses pengambilan gambar**

Gambar 4.5 menjelaskan bagaimana program tugas akhir ini melakukan perintah pengambilan gambar dengan menggunakan *library subprocess call* yang diimplementasikan kedalam *source code* berbahasa Python, proses pemanggilan diinisialisasikan kedalam *variable* bernama *grap\_cam*. Kemudian untuk format penamaan hasil foto diinisialisasikan dengan *variable img\_name*, untuk media penyimpanan diinisialisasikan dengan *variable img\_path*.

### **4.3.3. Implementasi Kamera dengan Sensor PIR**

Pengambilan gambar dengan menggunakan kamera yang terpasang pada Raspberry Pi dapat dilihat pada Gambar 4.6 seperti dibawah ini.

```
Import RPi.GPIO as io
Import subprocess as sb
pir_pin = 7
io.setup(pir_pin, io.IN)

current_state = 0
previous_state = 0
try :

    while io.input(pir_pin)==1:
        current_state = 0
        print "ready"
        while True :

            current_state = io.input(pir_pin)
            if current_state==1 and
                previous_state==0:
                grap_cam = sb.Popen("sudo fswebcam
                                    -p , shell=True)
                elif current_state==0 and
```

```

previous_state==0:
print "tidak aktif"

```

**Gambar 4.6 Contoh implementasi sensor PIR dengan modul kamera**

Pada Gambar 4.6 dijelaskan bagaimana implementasi menghubungkan sensor infra merah PIR dengan modul kamera dengan menggunakan bahasa pemrograman Python, yang perlu dilakukan pertama kali adalah membuat proses inisialisasi sensor infra merah PIR pada Raspberry Pi agar mengenali *socket pin* GPIO mana yang mengenali dan dapat mengirim sinyal digital dari sensor infra merah PIR ke Raspberry PI ketika menerima *trigger*.

Hasil dari *trigger* yang masuk kedalam Raspberry Pi akan menjadi sebuah *trigger* baru untuk mengaktifkan proses pengambilan gambar menggunakan kamera yang telah terhubung pada Raspberry Pi dengan menggunakan aplikasi Fsw webcam yang dipanggil dengan menggunakan bantuan *library subprocess call* yang terdapat pada bahasa pemrograman Python.

#### 4.3.3.1. Implementasi HOG Descriptor

Gambar 4.7 adalah potongan dari fungsi HOG Descriptor yang terdapat pada *library* Open-CV Python, disini penulis menggunakan fungsi HOGDetectMultiScale yang merupakan bagian dari HOG Descriptor yang berfungsi untuk mengenali object dengan *multi-scale window*. Ada beberapa argument-argumen yang ada pada fungsi, diantaranya yaitu: (10)

- `Img`
- `Storage`
- `Win_stride()`
- `Padding()`
- `Scale()`
- `Group_threshold()`

```

found=list(HOGDetectMultiScale(img, storage ,
win_stride=(8,8) , padding=(32,32) , scale =
1.05 , group_threshold=2))

```

**Gambar 4.7 Fungsi HOGDescriptor**



Dari fungsi *HOG Descriptor* di atas akan menghasilkan *output* berupa *list[x,y]* yang berisi koordinat sebuah objek, disini objek yang dikenali adalah manusia. Sedangkan untuk mengenali apakah itu manusia atau bukan, maka ditambahkanlah sebuah kondisi seperti pada Gambar 4.8. Kondisi yang dipakai adalah dengan cara mencocokkan dan mengecek isi dari *list output* fungsi proses *HOG Descriptor*, dengan kata lain jika pada *list output* terdapat isi atau koordinat maka dicetak “manusia” namun ketika hasil dari *list output* fungsi proses *HOG Descriptor* berisi *null* atau kosong maka dicetak “bukan manusia”.

```
if len(found) > 0 :
    print " manusia"
else :
    print " bukan manusia "
```

**Gambar 4.8 Kondisi pengecekan list output HOG Descriptor**

#### **4.3.4.Implementasi Pengenalan dan Pengecekan MAC *address client* dengan menggunakan *access point* Raspberry Pi**

Pada Gambar 4.9 di bawah dijelaskan bagaimana cara agar *access point* yang telah dibuat berbasis kan Raspberry Pi dapat mendeteksi MAC *address client* yang terhubung ke *access point*. Dengan menggunakan perintah program *shell command ARP* yang berfungsi untuk menampilkan semua informasi *client* yang terhubung ke dalam *network* (13).

```
Import subprocess as sb
dict = {'b4:f0:ab:77:ee:82' : 'dududh ' ,
        '14:74:11:b1:17:aa' : 'anak ayam' }
get_mac = sb.Popen( " arp -n | awk ' { print $3
} ' > hasil.txt " , shell=True)
f = open('hasil.txt', 'r+')
flg=1
for line in f :
    mac = []
    mac.append(line.rstrip('\n'))
```

```

for k in mac :
    if str(k) in dict.keys():
        listAnggota = dict[str(k)]
        print listAnggota

```

**Gambar 4.9 Pengecekan MAC address client**

Untuk mengambil *MAC address client* yang terhubung digunakanlah perintah *shell command AWK* yang berfungsi untuk mengambil data-data atau bagian-bagian penting yang kita inginkan berdasarkan baris hasil dari *output* perintah *shell command*.

Untuk menjalankan kedua *shell command* di atas digunakanlah *library subprocess call* yang ada pada bahasa pemrograman Python, hasil dari proses tersebut kemudian akan disimpan dalam sebuah *file text*. *File text* tersebut akan berisi hasil dari dua perintah *subprocess call* yang telah dilakukan sebelumnya, kemudian isi dari *file text* akan dibandingkan dan dicek dengan isi *list dictionary* yang ada pada *script* bahasa pemrograman Python yang terlebih dahulu diisikan dengan *MAC address client* sebagai *key\_value* dari *list dictionary* dan nama *client* sebagai *value* nya. Ketika *MAC address client* yang terhubung kedalam *access point* dikenali sebagai *key\_value* yang ada pada *list dictionary* maka akan ditampilkan pesan berupa nama dari pemilik *MAC address* tersebut.

#### **4.3.4. Implementasi Pengiriman Gambar Melalui E-mail**

Pada Gambar 4.10 adalah merupakan hasil dari implementasi notifikasi kirim gambar melalui *e-mail*, dengan menggunakan *library SMTP* pada Python, potongan kode untuk implementasi *e-mail* seperti pada Gambar 4.11 dibawah ini (14).

```

msgRoot = mm (' related ')
msgRoot [ ' Subject ' ] = ' Pesan Email '
msgRoot [ ' From ' ] = fr
msgRoot [ ' To ' ] = to

```

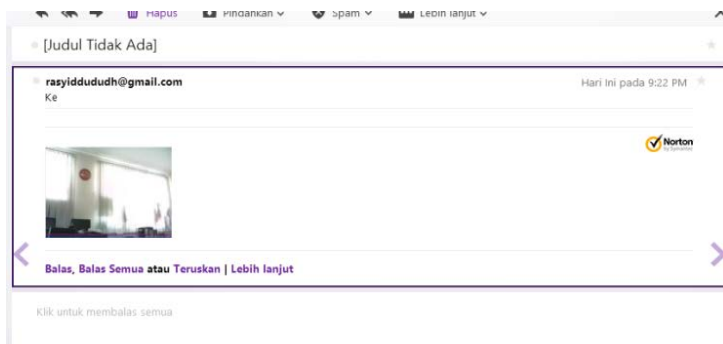
```

fp = open(image_path,'rb')
msgImage = mi(fp.read())
fp.close()
msgRoot.attach(msgImage)

smtp = smtplib.SMTP()
smtp.connect ('smtp.gmail.com')
smtp.starttls()
smtp.login('rasyiddududh@gmail.com',xxxxxxxxxx')
smtp.sendmail(fr,to,msgRoot.as_string())
smtp.quit()

```

**Gambar 4.10 Implementasi pengiriman gambar melalui e-mail**



**Gambar 4.0.11 Hasil pengiriman gambar ke e-mail**

#### 4.3.5. Implementasi Pengiriman Gambar Melalui TweetPony

Pengiriman notifikasi berupa *update status* Twitter dengan disertai gambar adalah salah satu fitur dari sistem keamanan cerdas ini. Karena pada saat ini kebanyakan orang menggunakan sosial media untuk melakukan interaksi dengan sesama, maka dari itu penulis menambahkan fitur kirim gambar melalui Twitter kedalam sistem keamanan cerdas ini.

Proses yang terjadi dalam pengiriman gambar melalui Twitter ini adalah ketika ada objek bergerak melintasi radius pancaran dari sinar infra merah PIR, maka *Wi-Fi chipset* yang

bertindak sebagai *access point* akan melakukan *scanning* dan pengecekan *MAC address client* yang terhubung.

Ketika hasil pengecekan tersebut di ketahui bahwa *MAC address client* tersebut terdapat dalam *list dictionary* Python maka akan menghasilkan sebuah *output* berupa nama dari pemilik atau anggota keluarga yang sebelumnya telah didaftarkan terlebih dahulu *MAC address*nya. Namun ketika tidak diketahui maka kamera akan menangkap gambar dari objek tersebut yang kemudian akan dilakukan proses pengenalan objek tersebut manusia atau bukan.

Potongan kode dari proses tersebut dapat di lihat pada Gambar 4.12

```

import Rpi.GPIO as io
import time
import subprocess as sb
import os
import tweetpony
import datetime
api=tweetpony.API(consumer_key="1zC07zMl7Sc98S5G
Un8Fi7ttn",consumer_secret="8TPzxAbvJmIfY9kXRtTB
GNMtdAlnAlyPrZyL8T0IoIngrVhQA",access_token="85
480692PENypuDv2aJZhDW9TlSVZxXXq9CVec64gzAMPCsNd"
,access_token_secret="YRRXp42np0svgQjXJfly0ksIRv
nWZxnQ9yNdFRWL8EH96")
today_date = datetime.datetime.today()
img_name = today_date.strftime('%m-%d-%y-%H%M')
img_path="/home/pi/anggota/{0}.jpg".format(img_n
ame)api.update_status_with_media(status = ("test
SPiCam Tugas Akhir " + listAnggota), media=
img_path)

```

**Gambar 4.12 Potongan Kode TweetPony**

Pada Gambar 4.12 di jelaskan bagaimana sistem dapat mengirim gambar melalui API yang dimiliki dari Twitter. Tetapi untuk menggunakan API tersebut pengguna harus memasukkan beberapa *parameter* yang harus dipenuhi, dimana pengguna dapat mendapatkan hak akses tersebut dengan cara mendaftarkan diri terlebih dahulu ke API Twitter.

Hak akses yang diberikan akan berupa *consumer\_key*, *consumer\_secret*, *access\_token*, dan *token\_secret*. Hasil dari implementasi proses ini dapat dilihat pada Gambar 4.13, dimana ketika melakukan implementasi penulis memberi kondisi dimana ada objek yang melintas dan kamera melakukan tangkap gambar. Hasil dari tangkap gambar diolah dengan OpenCV yang diketahui hasil dari proses tersebut adalah objek penyusup berupa manusia, yang kemudian akan diteruskan berupa notifikasi *update status* Twitter penulis.



**Gambar 4.13** Hasil dari implementasi TweetPony

#### 4.3.6. Implementasi ARP *Binding*

Penggunaan ARP *Binding* adalah untuk menangkal adanya serangan dengan menggunakan ARP *Spoofing* dan ARP *Poisoning* ke dalam jaringan yang telah dibangun.

ARP *Spoofing* dan ARP *Poisoning* dapat melakukan penggandaan MAC *address client* yang terhubung ke dalam jaringan. ARP *Binding* bekerja dengan cara melakukan *binding* atau pendaftaran IP *Address* beserta MAC *Address client* ke dalam ARP *Table* pada sistem jaringan.

Dengan cara tersebut maka tidak akan ada penggandaan atau *clone* dari IP *address* dan MAC *address* yang terhubung pada jaringan, ketika ada penggandaan IP *address* atau MAC *address* maka sistem akan otomatis melakukan drop paket data kepada client yang memiliki IP *address* atau MAC *address* yang

sama dengan *IP address* atau *MAC address* yang telah didaftarkan atau dibinding terlebih dahulu didalam *ARP Table*.

Konfigurasi *ARP Binding* dilakukan pada *terminal* Linux dengan menggunakan contoh *command shell* seperti pada Gambar 4.13 dibawah.

ARP	-s	172.63.0.6	14:74:11:B1:17:AA
-----	----	------------	-------------------

**Gambar 4.14 Penggunaan *ARP Binding***

## BAB V

### UJI COBA DAN EVALUASI

Pada bab ini akan dibahas mengenai uji coba dari segi fungsionalitas dan performa dari aplikasi. Uji coba fungsionalitas dan performa akan dibagi ke dalam beberapa skenario uji coba.

#### 5.1. Lingkungan Uji Coba

Dalam proses uji coba, dijelaskan mengenai gambaran lingkungan yang akan digunakan sebagai uji coba sistem. Uji coba akan dilakukan di dalam ruangan Laboratorium GCL jurusan Teknik Informatika ITS Surabaya, mini komputer Raspberry Pi dengan perangkat sensor akan ditempatkan pada salah satu sudut ruangan dekat dengan *server* dan akan menghadap langsung dengan pintu masuk yang bertujuan agar sensor dapat langsung mendeteksi pergerakan apabila ada seseorang atau objek yang masuk melalui pintu utama, jarak pintu utama dengan mini komputer Raspberry Pi sekitar 5 meter. Pengiriman hasil gambar hasil deteksi akan langsung dikirimkan ke *user* dan akan dilakukan *update status* Twitter *user* melalui jaringan internet LAN yang sudah terhubung dan sudah dikonfigurasi sebelumnya.



**Gambar 5.1 Foto Lingkungan Uji Coba**

Pada Gambar 5.1 digambarkan bagaimana situasi lokasi pengujian dengan sistem pengawas ruangan mini komputer

Raspberry Pi diletakkan pada salah satu sudut ruangan dekat dengan *server* dan menghadap ke arah pintu utama.

Lingkungan uji coba memiliki beberapa perangkat pendukung seperti berikut:

- Lingkungan ruangan di dalam Laboratorium GCL Teknik Informatika ITS
- Perangkat pengawas ruangan berupa mini komputer Raspberry Pi, sensor infra merah PIR, modul USB kamera Logitech C-170, *Wi-Fi chipset* TP-LINK-WN722N
- USB *Hub adaptor* dengan daya 5 V 1000Ma
- 1 buah komputer server
- Laptop Dell Inspiron N4010 Intel® Core™ i3CPU M350 @2,27 Ghz (4CPUs), ~ 2,3 Ghz
- PuTTY
- Tmux

## 5.2.Uji Coba Fungsionalitas Sistem

Uji coba fungsionalitas merupakan sebuah pengujian terhadap jalannya fungsi-fungsi utama yang ada pada aplikasi. Uji coba fungsionalitas pada perangkat sistem pengawas ruangan ini terdiri dari Raspberry Pi, sensor infra merah PIR, USB modul kamera meliputi, *Wi-Fi chipset*. Semua uji coba sistem ini dilakukan dengan cara melakukan *remote computer* dengan menggunakan aplikasi PuTTY yang ada pada laptop penulis, ada beberapa fungsionalitas utama yang ada dan diuji cobakan pada sistem ini yaitu:

1. Pendeteksian objek bergerak dengan sensor infra merah PIR.
2. Pendeteksian pengenalan MAC *address client* melalui *access point*.
3. Pengenalan objek manusia atau bukan
4. Pengambilan gambar dan pengiriman gambar melalui *e-mail*



### 5.2.1. Uji Coba Fungsionalitas Pendeteksian Objek Bergerak dengan Sensor Infra Merah PIR

Proses pendeteksian objek bergerak dengan sensor infra merah PIR merupakan proses utama dari keseluruhan sistem, karena proses ini adalah *trigger* untuk proses-proses selanjutnya yang ada pada sistem. Sensor PIR menerima sinyal dari gerakan dan suhu tubuh manusia atau objek bergerak yang berada dalam radius pancaran sensor infra merah PIR yang akan mengaktifkan fungsi-fungsi lain dari sistem ini. Proses uji coba sensor infra merah PIR ini akan diuji coba berdasarkan jarak antara sensor infra merah PIR dengan jarak manusia atau objek yang melintas disekitar radius pancaran sensor infra merah PIR, hasil uji coba dapat dilihat di Tabel 5.1.

**Tabel 5.1 Uji Coba Proses Sensor Infra Merah PIR**

No.	Jarak dengan objek	Status
1	1 meter	Berhasil
2	2 meter	Berhasil
3	3 meter	Berhasil
4	4 meter	Berhasil
5	5 meter	Berhasil

### 5.2.2. Uji Coba Fungsionalitas Proses Pengenalan MAC *address client* melalui *access point*

Proses pengenalan MAC *address client* merupakan lanjutan dari proses sebelumnya yaitu pendeteksian gerakan dengan sensor infra merah PIR, pada proses uji coba kali ini akan dilakukan dengan cara pengenalan MAC *address client* yang telah diisi sebelumnya terlebih dahulu kedalam sebuah *list dictionary* Python yang akan dibandingkan dengan hasil *subprocess call* ARP dengan range IP mulai dari 172.63.0.2 sampai dengan 172.63.0.10 (15), ada dua skenario pengujian fungsionalitas yang pertama adalah pengujian antara jarak *client* dengan *access point*, hasil uji coba dapat dilihat pada Tabel 5.2.

**Tabel 5.2 Uji coba jarak access point**

No.	Jarak <i>client</i>	Status
1	1 meter	Sukses
2	2 meter	Sukses
3	3 meter	Sukses
4	4 meter	Sukses
5	5 meter	Sukses

```

root@raspberrypi:/home/pi# python TA_FIX_1.py
pir module TEST (CTRL+C to exit)
waiting for PIR settle...
ready
tidak aktif
tidak aktif
tidak aktif
tidak aktif
motion detected
cek mac address
===== bukan anggota =====
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
Adjusting resolution from 480x640 to 640x480.
--- Capturing frame...
Captured frame in 0.00 seconds.
--- PROCESSING CAPTURED IMAGE...
Writing JPEG image to '/home/pi/penyusup/07-17-14-1405.jpg'.
bukan manusia

```

**Gambar 5.2 Kondisi dimana MAC *address client* tidak dikenali**

Skenario yang kedua adalah perbandingan berdasarkan MAC *address client* yang tersambung kedalam *access point* dengan daftar MAC *address client* yang telah diisikan terlebih dahulu kedalam *list dictionary* Python, apakah dapat dikenali atau tidak dikenali, hasil uji coba skenario kedua dapat dilihat pada Gambar 5.2. Pada Gambar 5.3 dijelaskan kondisi ketika MAC *address client* ada dan dikenali oleh sistem pengawas ruangan ini. Isi dari *list directory* Python adalah *key\_value* dan *value*, dimana - *key\_value* akan di isi oleh MAC *address client* yang telah di daftarkan sebelumnya atau dimasukkan sebelumnya di dalam *list directory* Python. Sedangkan *value* akan berisi nama dari pemilik MAC *address* tersebut, disini akan diisi nama penulis dan beberapa nama dari teman penulis yang mengikuti uji coba sistem keamanan ini.

```

172.63.0.6 dev wlan0 lladdr 14:74:11:b1:17:aa used 52/50/33 probes 1 STALE

*** Round 1, deleting 3 entries ***
10.151.63.166 dev eth0 lladdr b8:ac:6f:75:8c:83 ref 1 used 0/0/0 probes 4 REACHA
BLE

*** Round 2, deleting 1 entries ***
*** Flush is complete after 2 rounds ***
ready
tidak aktif
tidak aktif
tidak aktif
tidak aktif
tidak aktif
tidak aktif
tidak aktif
motion detected
cek mac address
welcome home Sindu
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
Adjusting resolution from 480x640 to 640x480.

```

**Gambar 5.3 Kondisi dimana MAC *address client* dikenali**

### 5.2.3. Uji Coba Notifikasi Twitter

Proses uji coba ini adalah langkah selanjutnya dari kedua uji coba sebelumnya yang telah dilakukan, merupakan kondisi dimana apa yang harus dilakukan oleh sistem ketika kondisi proses sebelumnya yaitu pengecekan *MAC address client* terpenuhi atau tidak. Notifikasi berupa *update status* pada Twitter penulis yang disertai dengan keterangan gambar dan keterangan apakah objek tersebut manusia atau bukan manusia. Pendeteksiab objek manusia atau bukan akan dilakukan dengan menggunakan HOGDescriptor yang telah diimplementasikan di dalam *source code* dengan bahasa pemrograman Python.

Pada Gambar 5.4 dijelaskan kondisi ketika *MAC address client* dikenali oleh sistem yang kemudian akan melakukan proses pengambilan gambar dan akan melakukan *update status* pada Twitter yang disertai dengan nama pemilik *MAC address* tersebut.

Seperti yang sudah dijelaskan sebelumnya API Twitter yang digunakan adalah TweetPony yang memiliki beberapa *parameter* yang harus dipenuhi oleh penggunaanya. Terdapat dua

hasil berbeda dari proses ini dimana hasil proses sangat ditentukan oleh proses pengecekan *MAC address client* yang terhubung ke dalam *access point* Raspberry Pi.



**Gambar 5.4 Kondisi dimana *MAC address client* terdeteksi dan dikenali sistem**



**Gambar 5.5 Kondisi ketika ada penyusup**

Pada Gambar 5.5 adalah ketika sistem tidak mengenali *MAC address client* yang kemudian akan terdeteksi sebagai penyusup, lalu sistem akan mengambil gambar dari objek tersebut yang kemudian dari hasil gambar tersebut, sistem akan mengenali apakah objek tersebut manusia atau bukan yang kemudian akan

diteruskan dengan notifikasi berupa update status Twitter penulis. Pada uji coba ini penyusup dikenali sebagai manusia.

### 5.3. Uji Coba Perangkat Sistem Utama

Pengujian perangkat sistem utama adalah serangkaian uji coba yang dilakukan oleh semua bagian sistem yang telah terhubung dan telah sesuai dengan urutan jalannya sistem dengan kondisi yang ada. Uji coba dimulai dengan pendeteksian gerakan dengan sensor infra merah PIR yang kemudian akan dilakukan pengecekan MAC *address client*, lalu setelah itu dilakukan proses pengambilan gambar dengan menggunakan kamera USB. Setelah itu hasil proses dari pengambilan gambar akan diketahui apakah objek tersebut manusia atau bukan dengan menggunakan Open-CV yang hasilnya akan dikirimkan berupa notifikasi berupa *e-mail* dan *update status* Twitter.

#### 5.3.1. Skenario Uji Coba 1

Skenario uji coba 1 adalah skenario uji coba perangkat sistem utama yang dilakukan di dalam Laboratorium GCL Teknik Informatika, dengan kondisi dalam Tabel 5.3.

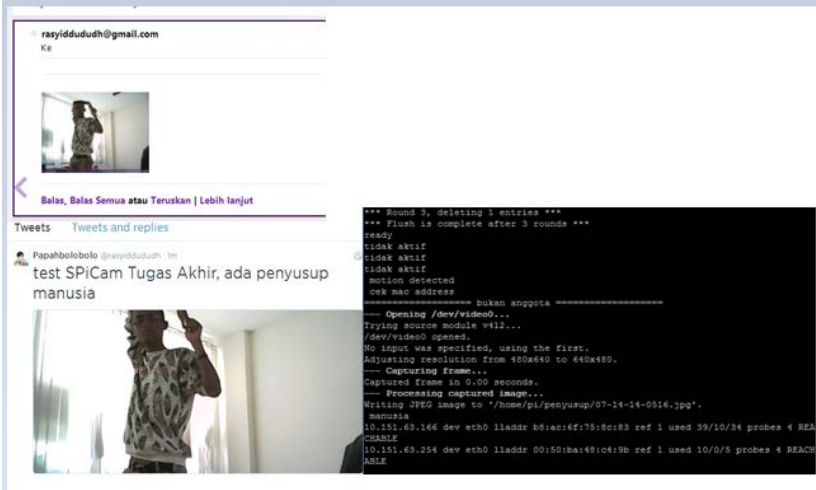
Pada Gambar 5.6 menunjukkan hasil dari kondisi skenario 1 yang ada seperti pada Tabel 5.3, kondisi yang disiapkan adalah ketika ada gerakan kemudian MAC *address client* tidak dikenali dan hasil dari objek pengambilan gambar menyatakan objek tersebut adalah manusia.

**Tabel 5.3 Tabel kondisi uji coba 1**

Sensor PIR	Bekerja
Objek	Manusia
Jarak	2 meter
Pengenalan MAC <i>address client</i>	Tidak dikenali

Sensor infra merah PIR akan bekerja ketika ada gerakan dan suhu tubuh manusia yang melintasi atau dalam radius pancarannya, sedangkan objek manusia adalah hasil dari proses pengolahan gambar dengan menggunakan Open-CV yang

menyatakan bahwa objek yang terdeteksi adalah manusia. Sedangkan *MAC address client* yang terhubung tidak terdapat atau terdaftar di dalam *list directory* Python.



Gambar 5.6 Hasil uji coba utama skenario 1

5.3.2.Skenario Uji Coba 2

Kondisi skenario uji coba 2 seperti pada Tabel 5.4. Pada kondisi kali ini, penulis memasukkan kondisi dimana ada hewan bergerak yang melintasi didepan sensor infra merah PIR yang kemudian oleh sistem akan dikenali sebagai “ bukan manusia” yang kemudian akan dikirmkan notifikasi, hasil seperti pada Gambar 5.7 di bawah ini.

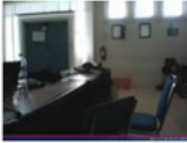
Tabel 5.4 Tabel Kondisi Uji Coba Skenario 2

Sensor PIR	Bekerja
Objek	Bukan manusia
Jarak	3 meter
Pengenalan <i>MAC address client</i>	Tidak dikenali

```

ready
tidak aktif
tidak aktif
tidak aktif
tidak aktif
tidak aktif
motion detected
cek mac address
===== bukan anggota =====
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
Adjusting resolution from 480x640 to 640x480.
--- Capturing frame...
Captured frame in 0.00 seconds.
--- Processing captured image...
Writing JPEG image to '/home/pi/penyusup/07-14-14-0547.jpg'.
bukan manusia
10.151.63.166 dev eth0 lladdr b8:ac:6f:75:8c:83 ref 1 used 95/11/6 probes 4 REACHABLE
10.151.63.254 dev eth0 lladdr 00:50:ba:48:c4:9b ref 1 used 8/0/7 probes 4 REACHABLE

```

rasyiddududh@gmail.com  
 Ke  


Balas, Balas Semua atau Teruskan | Lebih lanjut

Gambar 5.7 Hasil Uji Coba skenario 2

### 5.3.3. Uji Coba Skenario 3

Pada uji coba skenario 3 kali ini diberikan kondisi Tabel 5.5. Hasil dari uji coba skenario 3 dapat dilihat pada Gambar 5.8 dan 5.9. Gambar 5.8 adalah hasil dari proses pengambilan gambar dan pengecekan *MAC address client* yang dikenali sebagai “Sindu”, yang sebelumnya *MAC address client* tersebut sudah dimasukkan kedalam *list directory* pada *script* Python. *List directory* akan berisi *MAC address* sebagai *key* dan nama pemilik sebagai *value*. Gambar 5.9 adalah hasil uji coba skenario 3 yang menunjukkan bahwa hasil dari pengecekan *MAC address client* yang terhubung ke dalam *access point* dikenali oleh sistem.

rasyiddududh@gmail.com

Ke



**Balas, Balas Semua atau Teruskan | Lebih lanjut**

**Gambar 5.8 Notifikasi berupa e-mail**

```
172.63.0.6 dev wlan0 lladdr 14:74:11:b1:17:aa used 52/50/33 probes 1 STALE

*** Round 1, deleting 3 entries ***
10.151.63.166 dev eth0 lladdr b8:ac:6f:75:8c:83 ref 1 used 0/0/0 probes 4 REACHA
BLE

*** Round 2, deleting 1 entries ***
*** Flush is complete after 2 rounds ***
ready
tidak aktif
tidak aktif
tidak aktif
tidak aktif
tidak aktif
tidak aktif
tidak aktif
motion detected
cek mac address
welcome home Sindu
--- Opening /dev/video0...
Trying source module v4l2...
/dev/video0 opened.
No input was specified, using the first.
Adjusting resolution from 480x640 to 640x480.
```

**Gambar 5.9 Hasil Uji Coba Skenario 3**

**Tabel 5.5 Tabel Kondisi Uji Coba Skenario 3**

Sensor	Bekerja
Objek	Manusia
Jarak	4 meter



Pengenalan <i>client</i>	MAC <i>address</i>	Dikenali sebagai <i>Sindu</i>
--------------------------	--------------------	-------------------------------

## 5.4. Uji Coba Performa

Uji coba performa dilakukan agar dapat melihat dan memantau kinerja dari Raspberry Pi saat melakukan semua aktifitas dari sistem dan pada saat Raspberry Pi siaga.

### 5.4.1. Uji Coba Penggunaan CPU pada Raspberry Pi

Uji coba kali ini dilakukan untuk mengukur seberapa besar pemakaian kapasitas CPU pada Raspberry Pi pada saat Raspberry Pi sedang siaga maupun pada saat semua sistem berjalan pada Raspberry Pi. Ada 2 skenario dalam uji coba kali ini yaitu ketika *stand by* dan melakukan proses utama.

Pada Gambar 5.10 adalah kondisi CPU *usage* pada saat Raspberry Pi *stand by* atau tidak ada proses apapun yang berjalan, terlihat pemakaian CPU pada Raspberry Pi sangat kecil berbanding terbalik dengan saat Raspberry Pi melakukan proses utama, seperti pada Gambar 5.11. Tabel 5.6 menunjukkan perbandingan dari CPU Raspberry Pi.

```

root@raspberrypi:/home/pi# sar 1 5
Linux 3.10.36+ (raspberrypi) 17/07/14 _armv6l_ (1 CPU)

14:09:19      CPU      %user   %nice   %system   %iowait   %steal   %idle
14:09:20      all       1.16     0.00     1.16     0.00     0.00    97.67
14:09:21      all       1.14     0.00     3.41     0.00     0.00    95.45
14:09:22      all       2.27     0.00     1.14     0.00     0.00    96.59
14:09:23      all       1.10     0.00     6.59     0.00     0.00    92.31
14:09:24      all       1.09     0.00     7.61     0.00     0.00    91.30
Average:      all       1.35     0.00     4.04     0.00     0.00    94.61
root@raspberrypi:/home/pi# █

```

**Gambar 5.10 Kondisi CPU pada saat siaga**

```

root@raspberrypi:/home/pi# sar 1 5
Linux 3.10.36+ (raspberrypi) 17/07/14 _armv6l_ (1 CPU)

14:10:59      CPU      %user   %nice   %system   %iowait   %steal   %idle
14:11:00      all      94.00     0.00     6.00     0.00     0.00     0.00
14:11:01      all      93.00     0.00     7.00     0.00     0.00     0.00
14:11:02      all      97.00     0.00     3.00     0.00     0.00     0.00
14:11:03      all      98.00     0.00     2.00     0.00     0.00     0.00
14:11:04      all      88.00     0.00    12.00     0.00     0.00     0.00
Average:      all      94.00     0.00     6.00     0.00     0.00     0.00

```

**Gambar 5.11 Kondisi CPU pada saat melakukan proses utama**

**Tabel 5.6 Perbandingan penggunaan CPU**

No.	Penggunaan CPU ( %)	
	Siaga	Proses utama
1	1,16	94,00
2	1,14	93,00
3	2,27	97,00
4	1,10	98,00
5	1,09	88,00

#### 5.4.2. Uji Coba Penggunaan Memori pada Raspberry Pi

Pengujian penggunaan memori berguna untuk mengetahui seberapa besar memori yang digunakan pada saat Raspberry Pi melakukan proses utama dan pada saat Raspberry Pi *stand by*. Gambar 5.12 adalah memori pada saat *stand by*.

```
root@raspberrypi:/home/pi# free -t -m
              total        used         free       shared    buffers     cached
Mem:           437         112         325           0          13          66
-/+ buffers/cache:          32         404
Swap:           99           0           99
Total:          537         112         425
root@raspberrypi:/home/pi#
```

**Gambar 5.12 Penggunaan memori pada saat siaga**

Penggunaan memori pada saat *stand by* relative lebih kecil daripada saat menjalankan proses utama, seperti pada Gambar 5.13. Pada Tabel 5.6 menunjukkan perbandingan memori pada saat *stand by* dan pada saat melakukan proses utama. Tabel 5.6 menunjukkan perbandingan dari CPU Raspberry Pi.

```
root@raspberrypi:/home/pi# free -t -m
              total        used         free       shared    buffers     cached
Mem:           437         135         302           0          13          66
-/+ buffers/cache:          55         382
Swap:           99           0           99
Total:          537         135         402
```

**Gambar 5.13 Penggunaan memori pada saat proses utama**

**Tabel 5.7 Tabel perbandingan pemakaian memori**

No.	Pemakaian Memori (MB)	
	Pada saat siaga	Pada saat proses utama
1	112	135

*(Halaman ini sengaja dikosongkan)*

## **BAB VI PENUTUP**

Pada bab terakhir ini akan dibahas mengenai kesimpulan yang diperoleh selama pengerjaan Tugas Akhir hingga selesai. Pada bab ini juga akan menjawab pertanyaan yang dijabarkan pada BAB I. Untuk memperbaiki semua kelebihan dan kekurangan dari sistem, akan dijelaskan pada subbab saran.

### **6.1.Kesimpulan**

Berikut adalah beberapa kesimpulan yang diperoleh dari proses pengerjaan Tugas Akhir ini:

1. Perangkat ini dapat mendeteksi objek yang berjarak hingga 5 meter dari jarak sensor infra merah PIR.
2. Perangkat ini mudah untuk diaplikasikan di dalam ruangan khususnya pada rumah.
3. Dari segi biaya lebih murah daripada membangun sistem CCTV.
4. Perangkat ini memiliki kemampuan untuk mengenali anggota keluarga dengan cara mengenali *MAC address* yang terhubung ke *access point*.
5. Perangkat ini multifungsi, selain dapat menjadi pengawas ruangan juga dapat menjadi *access point*.

### **6.2.Saran**

Adapun saran dari penulis yang dapat diberikan dari kekurangan sistem yang ada, maupun pengembangan lebih lanjut yang dapat dilakukan yaitu:

1. Penambahan sensor aktif infra merah untuk dapat membedakan antara manusia dengan hewan.
2. Penambahan sistem alarm atau peringatan dini ketika objek penyusup tidak dikenali.
3. penambahan fitur *streaming* agar mudah untuk memantau lewat internet.

*(Halaman ini sengaja dikosongkan)*

## LAMPIRAN

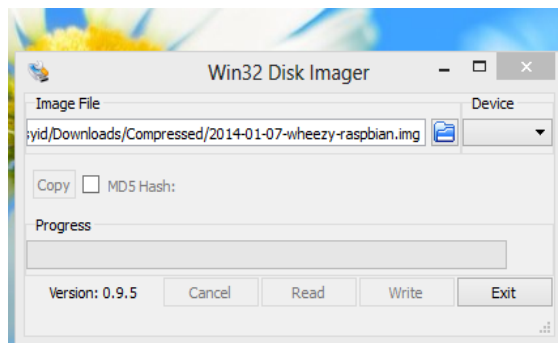
Bagian ini merupakan lampiran sebagai dokumen pelengkap dari buku Tugas Akhir dimana akan diberikan potongan kode sumber dari fungsi-fungsi yang digunakan untuk membangun sistem dan tabel-tabel yang berisikan hasil pengujian.

### A. Implementasi pada Raspberry Pi

#### 1. Menginstall Raspbian Wheezy

Proses instalasi sistem operasi pada Raspberry Pi adalah hal yang harus dilakukan dikarenakan Raspberry Pi seperti halnya suatu perangkat komputer perlu dipasang sistem operasi. Langkah- langkah untuk melakukan instalasi sistem operasi Raspbian Wheezy akan dijelaskan ke beberapa tahapan sebagai berikut:

- a) Siapkan sebuah kartu SD yang berukuran minimal 4GB.
- b) Unduh aplikasi *Win32 Disk Imager* untuk melakukan format kartu SD yang akan digunakan sebagai wadah *file system* operasi Raspbian Wheezy.



**Gambar A. 1 Aplikasi Win32 Disk Imager**

- c) Pada Gambar A. 1 ditunjukkan tampilan aplikasi *Win32 Disk Manager*, pilih *drive* kartu SD yang ingin di format

dan sorot berkas *image* dari sistem operasi Raspbian Wheezy.

- d) Klik *Write* dan tunggu sampai proses nya selesai dan kartu SD siap digunakan untuk menjalankan sistem operasi Raspberry Pi.

## 2. Memasang OpenCV

Langkah langkah *install library* OpenCV pada Raspberry Pi:

- a) Memastikan sistem operasi Raspbian telah *update*:

```
sudo apt-get update
sudo apt-get upgrade
```

**Gambar A.2 Perintah *Update* dan *Upgrade***

- b) *Install* dependensi

Pada Gambar A.3 adalah perintah *install* dependensi. Dependensi ini digunakan untuk melakukan *install* paket-paket dari *library* Open-CV

```
sudo apt-get -y install build-essential cmake cmake-curses-gui pkg-config
libpng12-0 libpng12-dev libpng++-dev libpng3 libpnglite-dev
zlib1g-dbg zlib1g zlib1g-dev pngtools libtiff4-dev libtiff4 libtiffxx0c2
libtiff-tools libeigen3-dev
```

**Gambar A.3 Perintah *Install* Dependensi**

Gambar A.4 adalah perintah lanjutan untuk melakukan *install* dependensi dari produk Open-CV. Dependensi ini digunakan untuk melakukan proses olah gambar dengan menggunakan bahasa pemrograman Python dan menggunakan *library* Open-CV.



```

sudo apt-get -y install libjpeg8
libjpeg8-dev libjpeg8-dbg libjpeg-progs
ffmpeg libavcodec-dev libavcodec53
libavformat53 libavformat-dev
libgstreamer0.10-0-dbg
libgstreamer0.10-0 libgstreamer0.10-dev
libxine1-ffmpeg libxine-dev libxine1-
bin libunicap2 libunicap2-dev swig
libv4l-0 libv4l-dev python-numpy
libpython2.6 python-dev python2.6-dev
libgtk2.0-dev

```

**Gambar A.4 Perintah *Install* Dependensi**

- c) Kemudian *download library* OpenCV dari URL <http://opencv.org/downloads.html> , kemudian di *unzip file* opencv-2.4.8.zip. Lalu buat folder *release* atau *build* dengan perintah *mkdir release* atau *build* kemudian masuk ke folder tersebut.
- d) Yang terakhir dalam proses instalasi ialah membuat perintah pada Gambar A.5:

```

make
sudo make install

```

**Gambar A.5 Perintah *Build* OpenCV**

3. *Install Kernel TP-Link TL-WN722N.*
  - a. *Edit /etc/apt/sources.list (16)* , penambahan isi pada *source.list* dapat menggunakan perintah *nano* atau *vim* pada *command shell*. Berikut adalah contoh dari penambahan isi dari *source.list* untuk dapat menggunakan *driver* dari *Wi-Fi chipset* TP-Link WN722N, isi dari *source.list* adalah berupa

halaman-halaman *repository* yang telah menyediakan *firmware* dari TP-Link WN722N.

```
deb
http://ftp.us.debian.org/debian/
squeeze main non-free
deb      http://security.debian.org/
squeeze/updates main non-free
deb
http://ftp.us.debian.org/debian/
squeeze-updates main non-free
deb
http://backports.debian.org/debian-
backports squeeze-backports main
non-free
```

**Gambar A. 6** Isi dari *source.list*

- b. Update *package-cache*.

```
sudo apt-get update
```

**Gambar A. 7** Perintah untuk *update*

- c. *Download firmware yang dibutuhkan*

```
sudo apt-get install firmware-atheros
sudo wget
http://linuxwireless.org/download/htc_
fw/1.3/htc_9271.fw
sudo cp htc_9271.fw /lib/firmware
```

**Gambar A. 8** *Install firmware*

- d. *Add adapter TP-Link TL-WN722N pada file konfigurasi.*

```
iface lo inet loopback
iface eth0 inet dhcp

# The wireless interface
auto wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa.conf
```

**Gambar A. 9** *file konfigurasi adapter*

Pada Gambar A.9 dijelaskan bagaimana isi dari *file* konfigurasi untuk menggunakan TP-Link TL-WN722N.

#### 4. *Install* API Twitter dengan menggunakan TweetPony

Untuk dapat menggunakan API Twitter penulis menggunakan API Twitter yang bernama TweetPony.

```
sudo pip install tweetpony
```

**Gambar A. 10 *Install* TweetPony**

#### 5. *Install* Hostapd

Untuk dapat mengubah Raspberry Pi menjadi *access point*, maka diperlukan sebuah aplikasi untuk menangani Raspberry Pi menjadi sebuah *router*. Maka digunakanlah aplikasi Hostapd, untuk proses *install* pertama-tama kita *download* terlebih dahulu *filenya* dengan menggunakan perintah *wget* seperti pada Gambar A. 11.

```
wget http://w1.fi/releases/hostapd-x.y.z.tar.gz
tar xzvf hostapd-x.y.z.tar.gz
cd hostapd-x.y.z/hostapd
```

**Gambar A. 11 *Download* Hostapd**

Kemudian kita perlu untuk mengkonfigurasi *file* Hostapd agar dapat menjalankan dan membaca *driver* yang *support* dengan nl80211. Dengan cara *copy file defconfig* ke dalam *file .config* lalu buka dengan *text editor*, dapat berupa *nano* atau *vim* seperti pada Gambar A.12.

```
cp defconfig .config
vi .config
```

**Gambar A. 12 *Copy file defconfig* ke dalam *file .config***

Setelah melakukan proses seperti pada Gambar A.12 kita lakukan langkah selanjutnya, temukan *line* pada *file .config* yang berisi seperti Gambar A.13.

```
#CONFIG_DRIVER_NL80211=y
```

**Gambar A. 13 Isi *file .config***

Hapus tanda pagar(#) agar Raspberry Pi dapat mengenali dan membaca *driver* dari nl80211.

## BIODATA PENULIS



**Rasyid Sindu Prihantono**, biasa dipanggil Rasyid atau Dududh, dilahirkan di Kediri pada tanggal 23 Oktober 1991. Penulis adalah anak kedua dari 2 bersaudara. Penulis telah menempuh pendidikan formal di SDK Santa Maria Kediri (1998-2004), SMP Negeri 1 Kediri (2004-2007), SMA Negeri 1 Kediri (2007-2010). Pada tahun 2010 penulis diterima di strata satu Jurusan Teknik Informatika Fakultas Teknologi

Informasi, Institut Teknologi Sepuluh Nopember Surabaya angkatan 2010 yang terdaftar dengan NRP. 5110100052. Di Jurusan Teknik Informatika ini, penulis mengambil bidang minat *Net Centric Computing* (NCC). Selama menempuh kuliah, penulis aktif sebagai staf departemen hubungan luar pada tahun kedua, dan pada tahun ketiga sebagai IC pengaderan departemen sumber daya manusia di Himpunan Mahasiswa Teknik Computer (HMTC), serta sebagai staf ahli departemen hubungan luar di BEM-FTiF. Pada beberapa acara kampus, penulis juga aktif menjadi panitia, baik sebagai anggota maupun koordinator. Selain itu penulis juga menjadi bagian *administrator* laboratorium GCL (*Grid Computing Laboratory*). Penulis dapat dihubungi melalui alamat *e-mail* di rasyid\_sindu@yahoo.com.